CLAIMS:

1. A method of performing a longest prefix match comprising the steps of:

a) filtering a key into a plurality of filter fields, each of which is associated with a respective filter table;

b) performing a longest prefix match (LPM) operation on each of the filter fields in their respective filter tables, wherein each LPM operation yields a result indicating lengths of prefixes potentially matching the key;

c) intersecting the results to obtain a set of potential prefix lengths; and

d) performing a series of hash lookups, based on the previously indicated potential prefix lengths, beginning with the longest potential prefix length and progressing to successively shorter potential prefix lengths until a matching prefix is found, which is the longest prefix matching the key.

2. The method as defined in claim 1 wherein each filter may use a different method for filter field extraction.

3. The method as defined in claim 1 wherein the key is an IP address.

4. The method as defined in claim 3 wherein the hash lookups are used for routing decisions.

5. The method as defined in claim 3 wherein the plurality of filter fields are created by extracting bits from the IP address.

6. The method as defined in claim 5 wherein the extracted bits are concatenated into filter fields, each filter field being a key for said associated filter table.

7. The method as defined in claim 1 wherein the longest prefix match operation on each filter field and associated filter table is performed in parallel.

8. The method as defined in claim 7 wherein a lookup table is used to perform the LPM operation.

9. The method as defined in claim 7 wherein a tree lookup is used to perform the LPM operation.

10. The method as defined in claim 1 applied to performing a new route insertion into a hash table.

11. The method as defined in claim 1 applied to the deletion of a route from a hash table.

12. The method as defined in claim 11 wherein the hash table is updated to delete the route.

13. The method as defined in claim 3 wherein only a portion of the IP address is filtered into filter fields.

14. The method as defined in claim 3 wherein the IP address is 128 bits long.

15. The method as defined in claim 3 wherein the IP address is 32 bits long.

16. The method as defined in claim 3 wherein an ideal offset filter is used.

17. A system for performing a longest prefix match, comprising:

a plurality of filter fields each created by filtering a key, each filter field being associated with a respecting filter table;

means to perform a longest prefix match (LPM) operation on each of the filter fields in their respective filter tables, wherein each LPM operation yields a result indicating potential prefix lengths of a longest prefix matching the key;

means to intersect the results to obtain a set of potential prefix lengths; and

means to search multiple logical hash tables, each associated with a specific prefix length, for a longest prefix match, in a linear fashion, wherein the search is performed using the set of potential prefix lengths starting at the longest potential prefix length and progressing to successively shorter potential prefix lengths until a matching prefix is found.

18. The system as defined in claim 17 wherein the filter fields are bit interleaved filters.

19. The system as defined in claim 17 wherein the key is an IP address.

20. The system as defined in claim 17 wherein the LPM operation is performed in a lookup table.

21. The system as defined in claim 19 wherein the IP address is 128 bits long consistent with IPv6.

22. The system as defined in claim 19 wherein the IP address is 32 bits long consistent with IPv4.